

BANCO DE DADOS ATUALIZADO

-- Criação do banco de dados

```
CREATE DATABASE IF NOT EXISTS sistema_profissionais_saude  
CHARACTER SET utf8mb4  
COLLATE utf8mb4_unicode_ci;
```

USE sistema_profissionais_saude;

-- Tabela de profissionais (tabela principal)

```
CREATE TABLE IF NOT EXISTS profissionais (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    cpf VARCHAR(14) NOT NULL UNIQUE,  
    data_nascimento DATE NOT NULL,  
    telefone VARCHAR(15) NOT NULL,  
    profissao VARCHAR(50) NOT NULL,  
    especialidade VARCHAR(50),  
    registro_profissional VARCHAR(30) NOT NULL,  
    uf_registro CHAR(2) NOT NULL,  
    servicos TEXT,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    senha VARCHAR(255) NOT NULL,  
    rg_arquivo VARCHAR(255),  
    cpf_arquivo VARCHAR(255),  
    registro_arquivo VARCHAR(255),  
    residencia_arquivo VARCHAR(255),
```

```
curriculum_arquivo VARCHAR(255),
ativo BOOLEAN DEFAULT TRUE,
data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
data_atualizacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
INDEX idx_profissao (profissao),
INDEX idx_especialidade (especialidade),
INDEX idx_uf_registro (uf_registro),
INDEX idx_email (email)
) ENGINE=InnoDB;
```

-- Tabela de auditoria (para registrar alterações importantes)

```
CREATE TABLE IF NOT EXISTS auditoria (
    id INT AUTO_INCREMENT PRIMARY KEY,
    tabela VARCHAR(50) NOT NULL,
    acao VARCHAR(20) NOT NULL COMMENT 'INSERT, UPDATE, DELETE',
    registro_id INT NOT NULL,
    dados_anteriores TEXT,
    usuario VARCHAR(100),
    ip VARCHAR(45),
    data_hora TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    INDEX idx_tabela (tabela),
    INDEX idx_acao (acao),
    INDEX idx_registro_id (registro_id)
) ENGINE=InnoDB;
```

-- Tabela de sessões (para controle de login)

```
CREATE TABLE IF NOT EXISTS sessoes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    profissional_id INT NOT NULL,  
    token VARCHAR(255) NOT NULL,  
    data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    data_expiracao TIMESTAMP NOT NULL,  
    ip VARCHAR(45),  
    user_agent VARCHAR(255),  
    FOREIGN KEY (profissional_id) REFERENCES profissionais(id) ON DELETE  
    CASCADE,  
    INDEX idx_token (token),  
    INDEX idx_profissional_id (profissional_id)  
) ENGINE=InnoDB;
```

-- Tabela de serviços (para normalização)

```
CREATE TABLE IF NOT EXISTS servicos (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL UNIQUE,  
    descricao TEXT,  
    ativo BOOLEAN DEFAULT TRUE  
) ENGINE=InnoDB;
```

```
-- Tabela de relação profissional_servico (many-to-many)

CREATE TABLE IF NOT EXISTS profissional_servico (

    profissional_id INT NOT NULL,

    servico_id INT NOT NULL,

    PRIMARY KEY (profissional_id, servico_id),

    FOREIGN KEY (profissional_id) REFERENCES profissionais(id) ON DELETE CASCADE,

    FOREIGN KEY (servico_id) REFERENCES servicos(id) ON DELETE CASCADE

) ENGINE=InnoDB;
```

```
-- Inserção de serviços padrão

INSERT INTO servicos (nome, descricao) VALUES

('Consulta', 'Consulta médica ou de outro profissional'),

('Exame', 'Realização de exames diversos'),

('Procedimento', 'Procedimentos clínicos ou terapêuticos'),

('Orientação', 'Orientação e aconselhamento'),

('Terapia', 'Sessões de terapia'),

('Acompanhamento', 'Acompanhamento contínuo do paciente'),

('Atividade Física Adaptada', 'Exercícios físicos adaptados para necessidades especiais'),

('Psicopedagogia', 'Acompanhamento psicopedagógico'),

('Fonoaudiologia', 'Terapia fonoaudiológica'),

('Terapia Ocupacional', 'Terapia ocupacional'),

('Esporte e Lazer Inclusivo', 'Atividades esportivas e de lazer inclusivas');
```

-- Trigger para auditoria de profissionais (INSERT)

DELIMITER //

CREATE TRIGGER after_profissionais_insert

AFTER INSERT ON profissionais

FOR EACH ROW

BEGIN

INSERT INTO auditoria (tabela, acao, registro_id, usuario, ip)

VALUES ('profissionais', 'INSERT', NEW.id, CURRENT_USER(),
CONNECTION_ID());

END//

DELIMITER ;

-- Trigger para auditoria de profissionais (UPDATE)

DELIMITER //

CREATE TRIGGER before_profissionais_update

BEFORE UPDATE ON profissionais

FOR EACH ROW

BEGIN

DECLARE dados_anteriores TEXT;

SET dados_anteriores = CONCAT(

'nome=', OLD.nome, ';',

'email=', OLD.email, ';',

'profissao=', OLD.profissao, ';',

'especialidade=', IFNULL(OLD.especialidade, 'NULL'), ';',

'registro_profissional=', OLD.registro_profissional, ';',

'ativo=', OLD.ativo

);

```
INSERT INTO auditoria (tabela, acao, registro_id, dados_antiores, usuario, ip)

VALUES ('profissionais', 'UPDATE', OLD.id, dados_antiores,
CURRENT_USER(), CONNECTION_ID());

END//

DELIMITER ;
```

-- Trigger para auditoria de profissionais (DELETE)

```
DELIMITER //
```

```
CREATE TRIGGER before_profissionais_delete
```

```
BEFORE DELETE ON profissionais
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    DECLARE dados_antiores TEXT;
```

```
    SET dados_antiores = CONCAT(
```

```
        'nome=', OLD.nome, ';',
```

```
        'email=', OLD.email, ';',
```

```
        'profissao=', OLD.profissao, ';',
```

```
        'especialidade=', IFNULL(OLD.especialidade, 'NULL'), ';',
```

```
        'registro_profissional=', OLD.registro_profissional
```

```
    );
```

```
INSERT INTO auditoria (tabela, acao, registro_id, dados_antiores, usuario, ip)
```

```
VALUES ('profissionais', 'DELETE', OLD.id, dados_antiores,
CURRENT_USER(), CONNECTION_ID());
```

```
END//
```

```
DELIMITER ;
```

-- View para listagem simplificada de profissionais

CREATE VIEW vw_profissionais AS

SELECT

p.id,

p.nome,

p.profissao,

p.especialidade,

p.registro_profissional,

p.uf_registro,

p.email,

p.ativo,

GROUP_CONCAT(s.nome SEPARATOR ', ') AS servicos_oferecidos,

p.data_cadastro,

p.data_atualizacao

FROM

profissionais p

LEFT JOIN

profissional_servico ps ON p.id = ps.profissional_id

LEFT JOIN

servicos s ON ps.servico_id = s.id

GROUP BY

p.id;

SHOW DATABASES LIKE 'sistema_profissionais_saude';

```
USE sistema_profissionais_saude;
```

```
SHOW TABLES;
```

```
SELECT cpf, email FROM profissionais WHERE email = 'email_inserido_no_login';
```

```
SELECT cpf FROM profissionais LIMIT 5;
```

```
ALTER TABLE profissionais
```

```
ADD COLUMN grau_autismo VARCHAR(20) COMMENT 'Leve/Moderado/Severo',
```

```
ADD COLUMN esportes_recomendados TEXT;
```

```
-- adicionada esta linha para armazenar foto da pessoa
```

```
ALTER TABLE profissionais ADD COLUMN foto VARCHAR(100) DEFAULT NULL  
COMMENT 'Caminho da foto do profissional';
```

```
select * from profissionais
```

```
CREATE TABLE IF NOT EXISTS agendamentos (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    profissional_id INT NOT NULL,
```

```
    nome_responsavel VARCHAR(100) NOT NULL,
```

```
    nome_crianca VARCHAR(100),
```

```
    grau_autismo_crianca VARCHAR(20),
```

```
    contato_responsavel VARCHAR(30),
```

```
    data_agendamento DATE NOT NULL,
```

```
    horario TIME NOT NULL,
```

```
    observacoes TEXT,
```



```
status ENUM('pendente', 'confirmado', 'cancelado') DEFAULT 'pendente',
data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

FOREIGN KEY (profissional_id) REFERENCES profissionais(id) ON DELETE
CASCADE,

INDEX idx_profissional (profissional_id),
INDEX idx_data_hora (data_agendamento, horario)
) ENGINE=InnoDB;
```

-- Categorias do blog

```
CREATE TABLE IF NOT EXISTS categorias (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(50) NOT NULL UNIQUE
);
```

-- Autores do blog (podem ser os próprios profissionais ou admins)

```
CREATE TABLE IF NOT EXISTS autores (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    perfil VARCHAR(50) DEFAULT 'editor' -- ex: admin, editor, profissional
);
```

-- Posts do blog

```
CREATE TABLE IF NOT EXISTS posts (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    autor_id INT NOT NULL,  
    categoria_id INT NOT NULL,  
    titulo VARCHAR(255) NOT NULL,  
    resumo TEXT,  
    conteudo LONGTEXT NOT NULL,  
    imagem_capa VARCHAR(255),  
    data_publicacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    publicado BOOLEAN DEFAULT TRUE,  
    FOREIGN KEY (autor_id) REFERENCES autores(id) ON DELETE CASCADE,  
    FOREIGN KEY (categoria_id) REFERENCES categorias(id) ON DELETE SET  
    NULL  
);
```

```

INSERT INTO posts (categoria_id, titulo, resumo, conteudo, data_publicacao,
publicado) VALUES
(
1,
'Terapias Eficazes para Crianças Autistas',
'Conheça as principais terapias que ajudam no desenvolvimento de crianças com
autismo, incluindo ABA, terapia ocupacional e fonoaudiologia.',
'O autismo é um transtorno do neurodesenvolvimento que afeta a comunicação e o
comportamento. As terapias mais recomendadas incluem:\n\n- ABA (Análise do
Comportamento Aplicada)\n- Terapia Ocupacional\n- Fonoaudiologia\n\nCada terapia
deve ser adaptada às necessidades individuais da criança para melhores resultados.',
'2025-06-17 10:30:00',
1
);
SELECT * FROM autores WHERE id = 1;

```

```

CREATE TABLE IF NOT EXISTS comentarios (
id INT AUTO_INCREMENT PRIMARY KEY,
post_id INT NOT NULL,
nome VARCHAR(100) NOT NULL,
email VARCHAR(100) NOT NULL,
comentario TEXT NOT NULL,
data_publicacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
aprovado BOOLEAN DEFAULT FALSE,
FOREIGN KEY (post_id) REFERENCES posts(id) ON DELETE CASCADE
);

```

```
CREATE TABLE IF NOT EXISTS usuarios (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    senha VARCHAR(255) NOT NULL,  
    perfil ENUM('admin', 'editor') DEFAULT 'editor',  
    criado_em TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE categorias_post01 (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL UNIQUE  
);
```

```
SHOW TABLES;
```

```
select * from usuarios
```